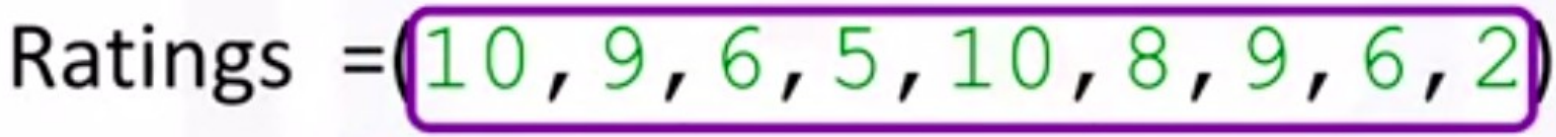
WEEK 2

Tuples

- Tuples are an ordered sequence
- Here is a Tuple “Ratings”
- Tuples are written as comma-separated elements within parentheses



These are values inside the parentheses. In Python, there are different types:

Tuples

```
Tuple1 =("disco",10,1.2)
```

0	"disco"
1	10
2	1.2

Tuple1[0]: "disco"

Tuple1[1]: 10

Tuple1[2]: 1.2

We can also access the last element. In Python, we can use negative index.

Tuples

```
Tuple1 =("disco",10,1.2)
```

-3	0	"disco"	Tuple1[-3]: "disco"
-2	1	10	Tuple1[-2]: 10
-1	2	1.2	Tuple1[-1]: 1.2

The relationship is as follows. The corresponding values are shown here.

Question

Consider the following tuple:

```
1  say_what=('say',' what', 'you', 'will')
```

what is the result of the following **say_what[-1]**

- 'say'
- ' what'
- 'you'
- 'will'

Correct

Correct. An index of -1 corresponds to the last index of the tuple, in this case, the string 'will'.

Skip Continue

Tuples

("disco", 10, 1.2)

tuple2 = tuple1 + ("hard rock", 10)

("disco", 10, 1.2, "hard rock", 10)

0	1	2	3	4

The result is the following with the following index.

Question

Consider the following tuple **A=(1,2,3,4,5)**, what is the result of the following:**A[1:4]**:

- (2, 3, 4)
- (3, 4,5)
- (2, 3, 4,5)

Correct
Correct. These indexes correspond to **elements** 1,2 and 3 of the tuple.

Skip Continue

Tuples: Slicing

("disco", 10, 1.2, "hard rock", 10)

tuple2[3:5]:("hard rock", 10)

we use the following command. Notice, how the last index is one larger than the

Question

Consider the following tuple **A=(1,2,3,4,5)**, what is the result of the following: **len(A)**

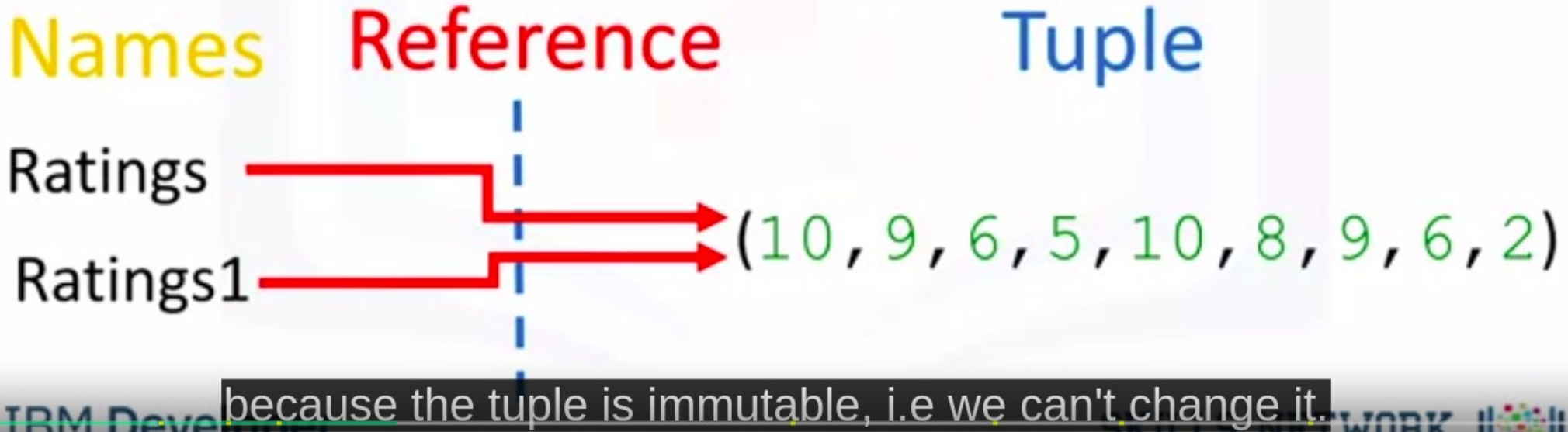
- 4
- 5
- 6

Correct

Correct. The function **len** returns the number of items of a tuple.

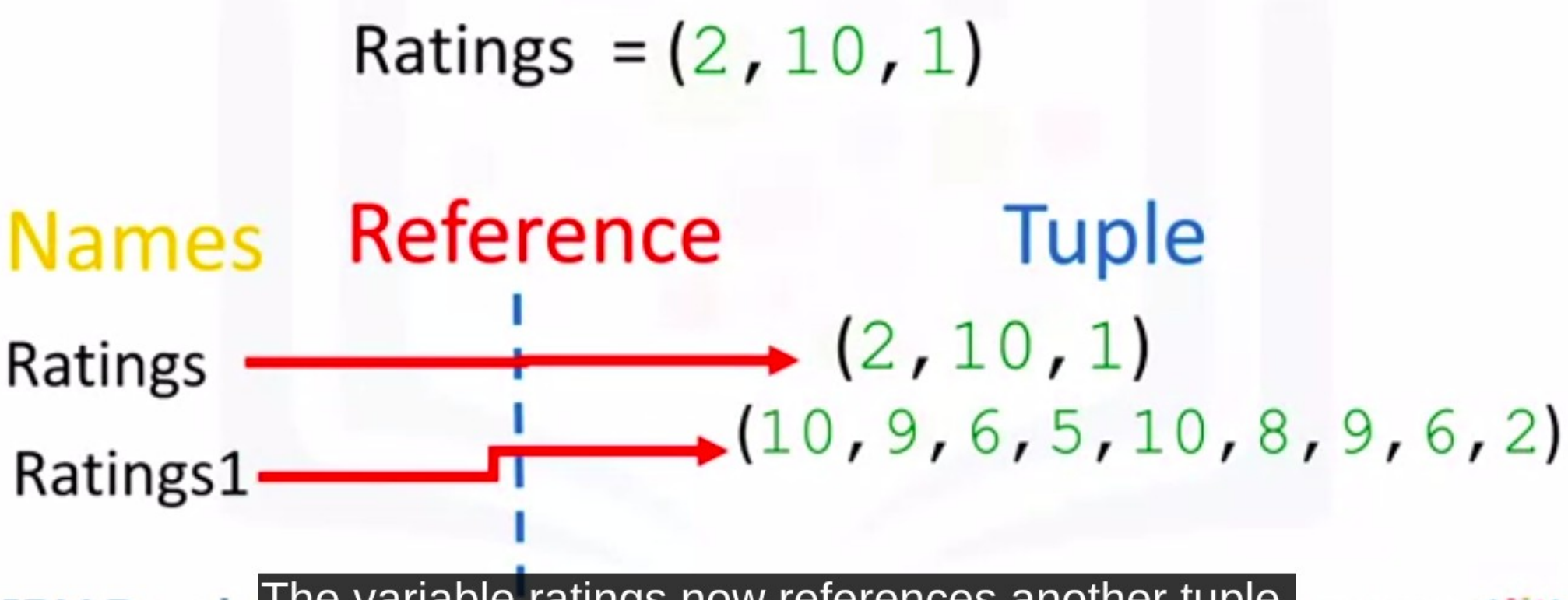
Skip Continue

Tuples: Immutable



because the tuple is immutable, i.e we can't change it.

Tuples: Immutable



The variable ratings now references another tuple.

Tuples: Immutable

```
Ratings =(10,9,6,5,10,8,9,6,2)

RatingsSorted=sorted(Ratings)

[2, 5, 6, 6, 8, 9, 9, 10, 10]
```

list. For more on functions, see our video on functions.

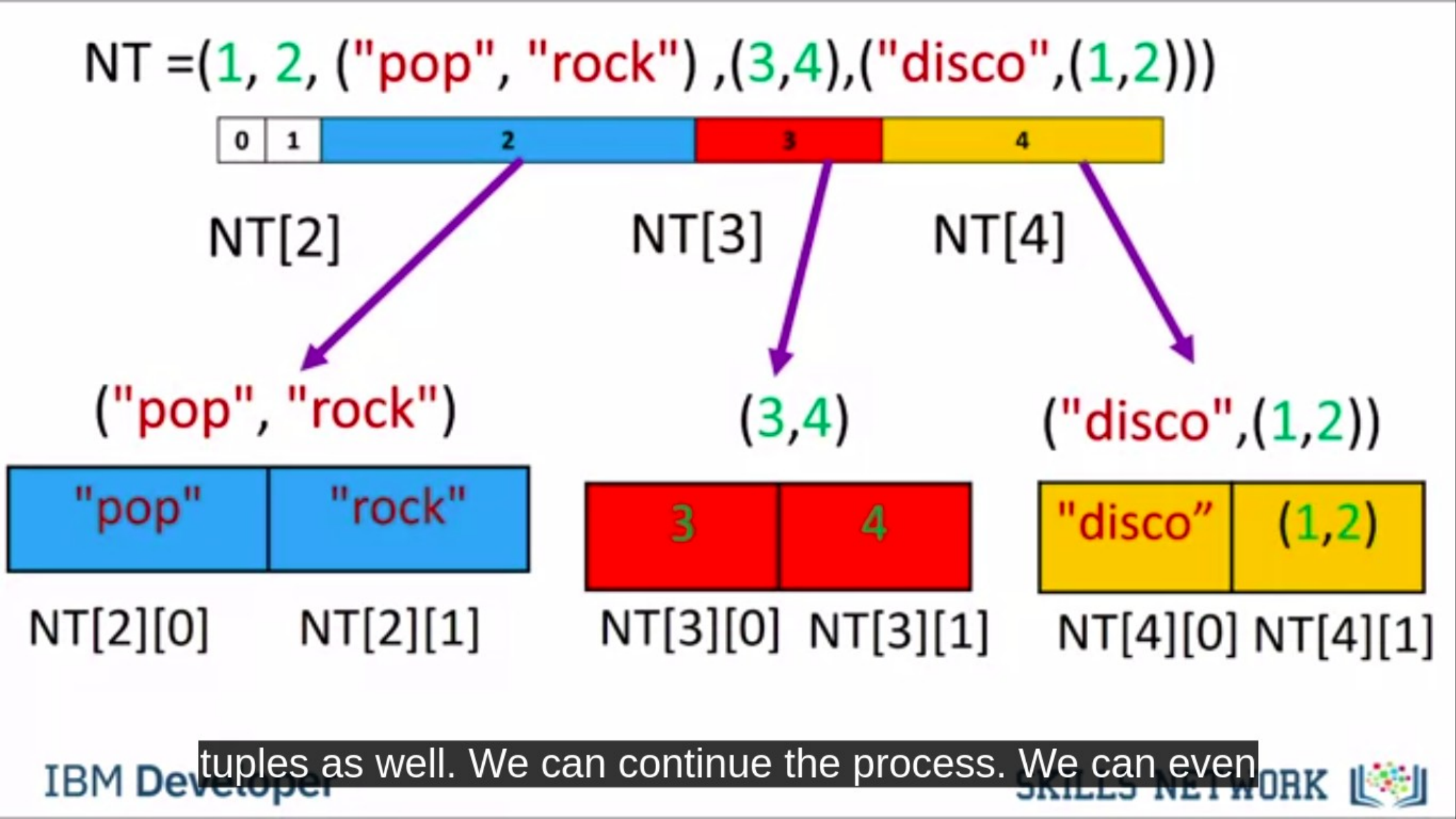
Tuples: Nesting

NT =(1, 2, ("pop", "rock") ,(3,4),("disco",(1,2)))

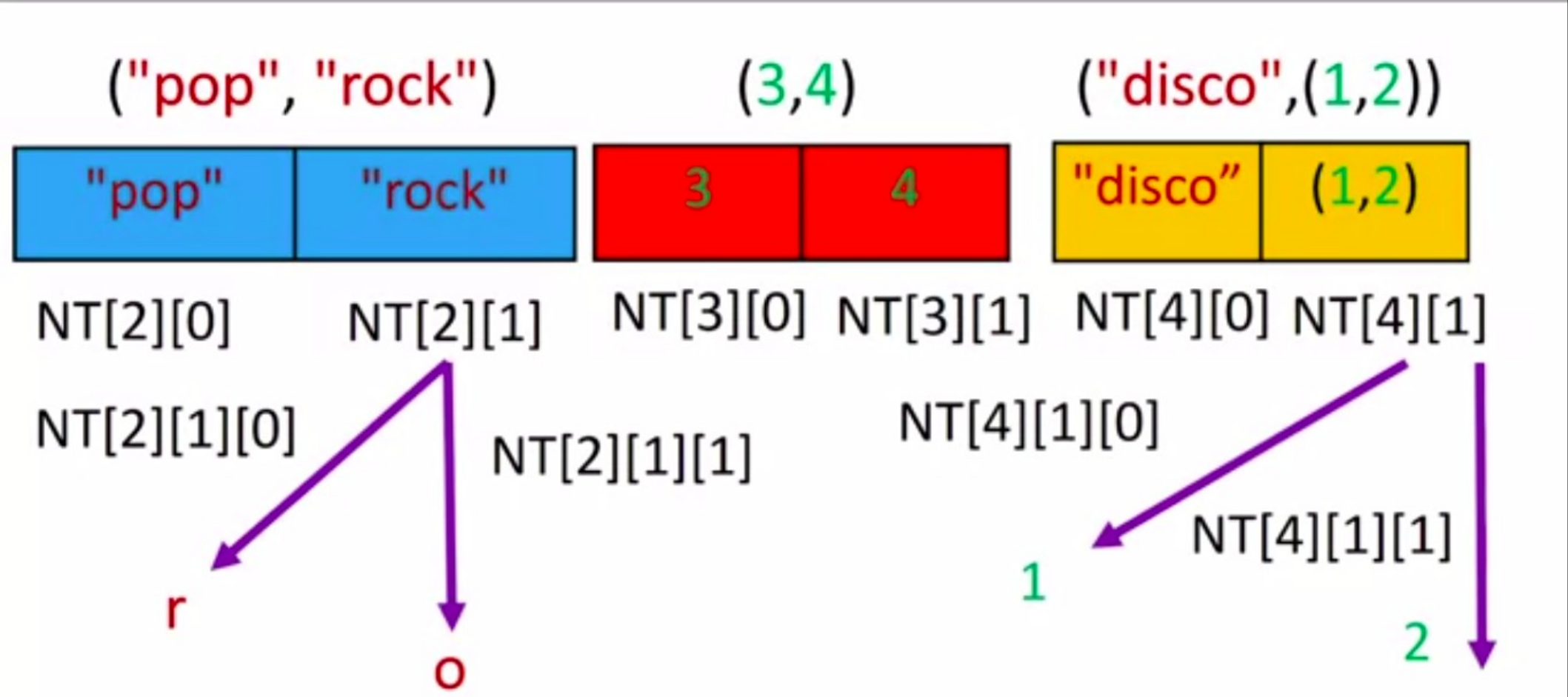
NT[2]: ("pop", "rock") [1] =“rock”

NT[2] [1] =“rock”

the tuple variable NT. It is helpful to visualize this as a



tuples as well. We can continue the process. We can even



We can access different characters in the string or various elements in the

Lists

- Lists are also ordered sequences
- Here is a List “L”
- A List is represented with square brackets
- List mutable

```
L =["Michael Jackson", 10.1,1982]
```

In many respect, lists are like tuples. One key difference is they are mutable.

Question

Consider the following list **B=[1,2,[3,'a'],[4,'b']]**, what is the result of the following:**B[3][1]**

- [4,"b"]
- "b"
- "c"

Correct
Correct.

Skip Continue

Lists

```
L =["Michael Jackson", 10.1,1982]
```

0	"Michael Jackson"
1	10.1
2	1982

L[0]: "Michael Jackson"

L[1]: 10.1

L[2]: 1982

We can also access the last element. In Python, we can use a negative index;

Lists

```
L =["Michael Jackson", 10.1,1982]
```

-3	0	"Michael Jackson"	L[-3]: "Michael Jackson"
-2	1	10.1	L[-2]: 10.1
-1	2	1982	L[-1]: 1982

are as follows. We can also perform slicing in lists. For example, if we want

Lists: Slicing

L =["Michael Jackson", 10.1,1982, “MJ”, 1]

L[3:5]:[“MJ”, 1]

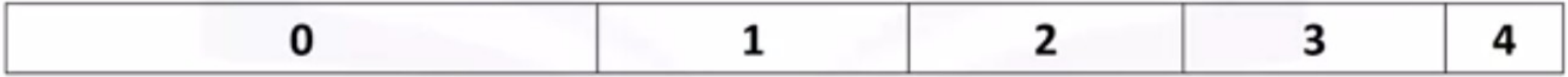
Notice how the last index is one larger than the length of the list.

Lists

```
L=["Michael Jackson", 10.1,1982]
```

```
L1 = L+["pop",10]
```

```
L1=["Michael Jackson", 10.1,1982, "pop",10]
```



them. The result is the following. The new list

Question

What is the result of the following operation

```
1  [1,2,3]+[1,1,1]
```

- [2,3,4]
- [1, 2, 3, 1, 1, 1]
- TypeError

Correct

Correct. The addition operator corresponds to concatenating a list.

Skip Continue

Lists

```
L=["Michael Jackson", 10.1,1982]

L.extend(["pop",10])

L=["Michael Jackson", 10.1,1982, "pop",10]
```

0	1	2	3	4

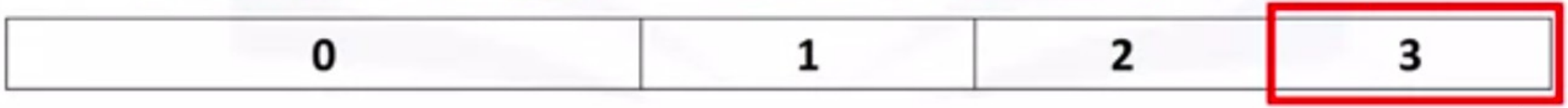
"L1," the original list, "L," is modified by adding two new elements.

Lists

```
L=["Michael Jackson", 10.1,1982]
```

```
L.append ([“pop”,10])
```

```
L=[“Michael Jackson”, 10.1,1982, [“pop”,10]]
```



there is only one more element. Index 3 contains the list we appended.

Question

What is the length of the list **A = [1]** after the following operation: **A.append([2,3,4,5])**

- ○ 5
- ○ 6
- ◉ 2

✓ **Correct**
Correct. Append only adds one element to the list .

Skip Continue

Lists

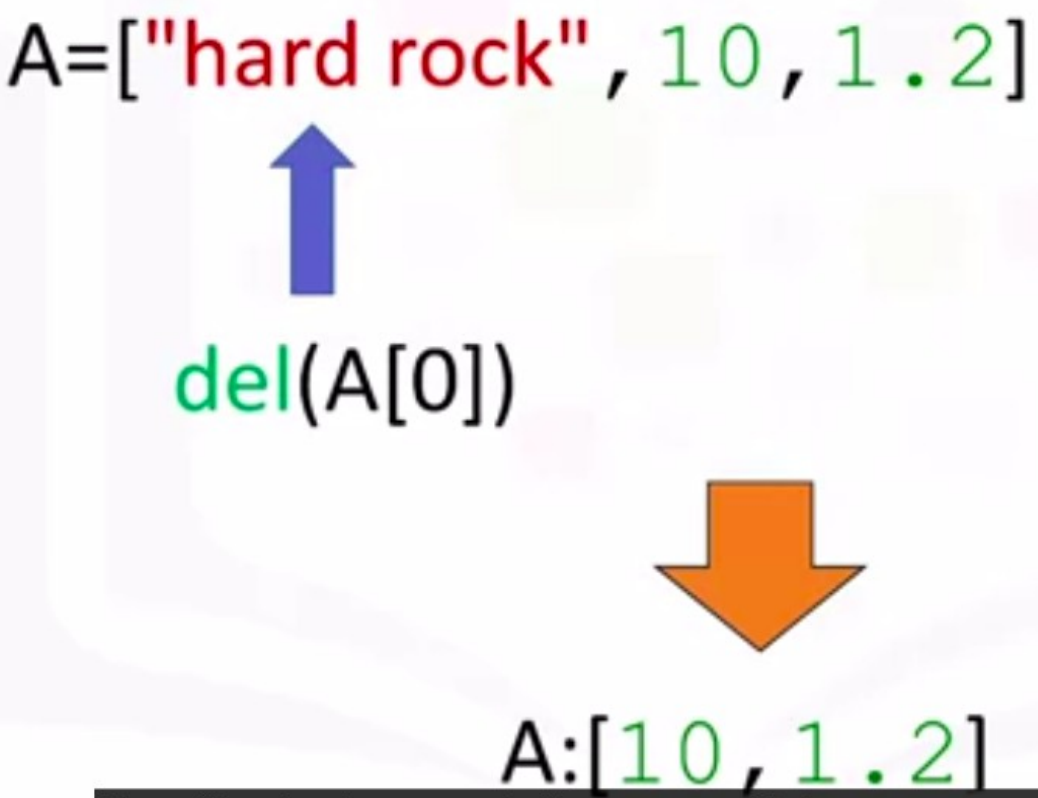
```
A=[“disco”,10,1.2]

A[0]=“hard rock”

A=["hard rock",10,1.2]
```

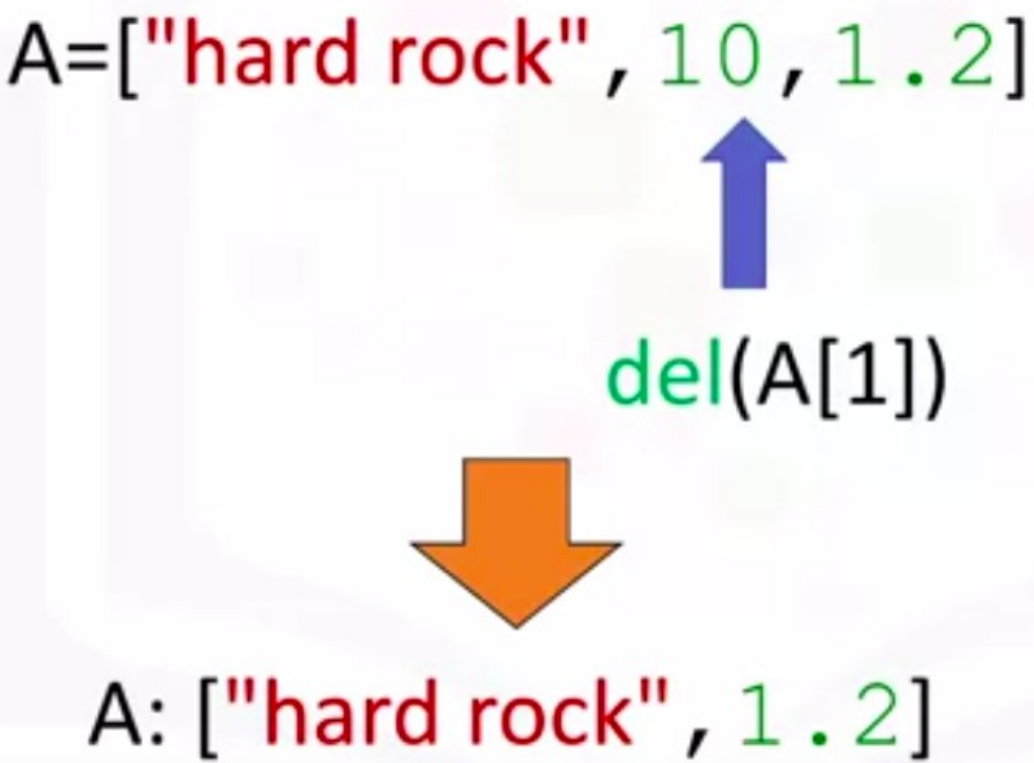
The list now becomes hard rock 10 1.2. We can delete an element of a list.

Lists



the first element the result becomes 10 1.2.

Lists



the list. We can convert a string to a list using

Question

What is the result of the following: **"Hello Mike".split()**

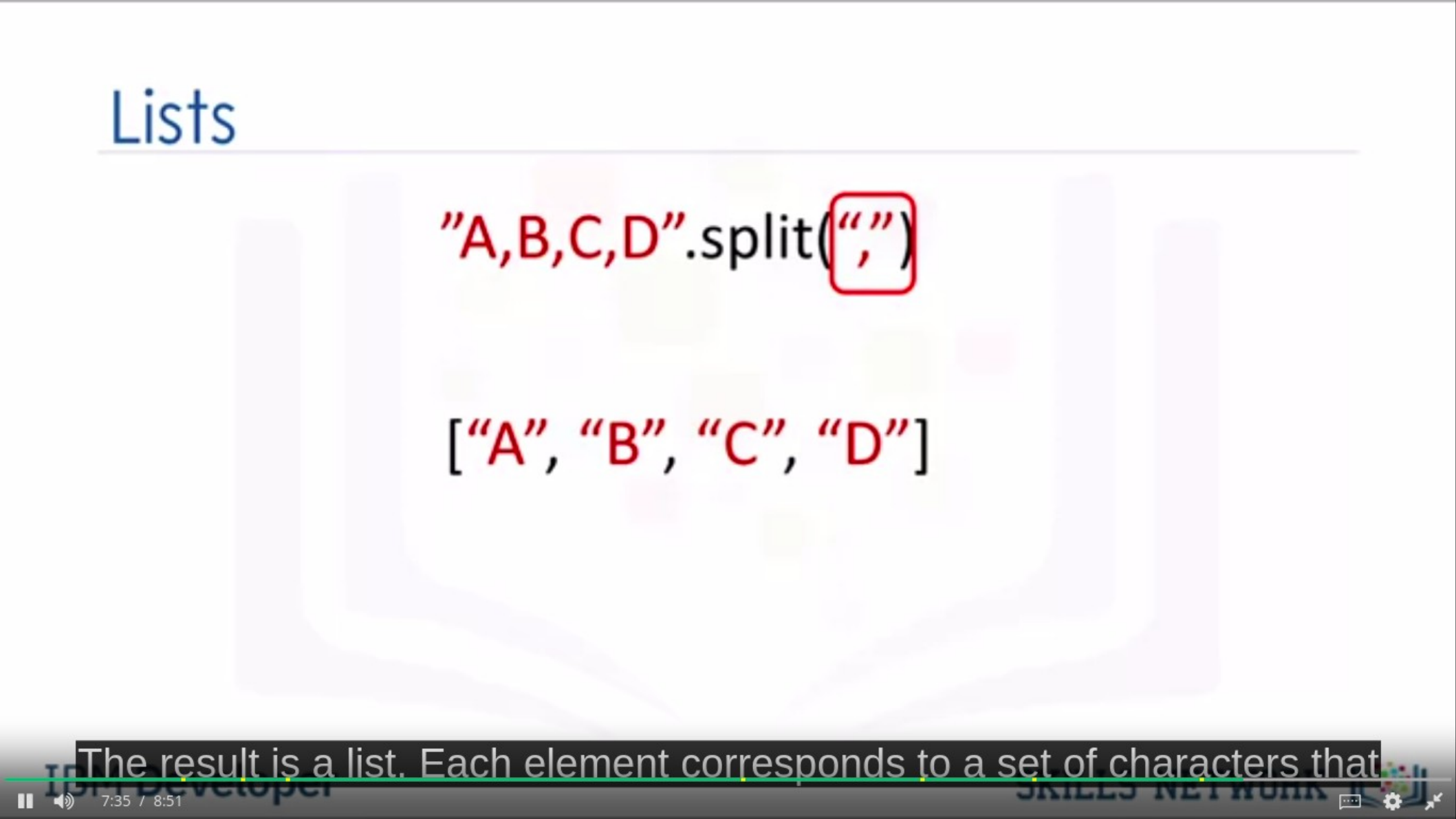
- ["HelloMike"]
- ["Hello","Mike"]
- ["H"]

Correct

Correct. The method split separates a string into a list based on the argument. If there is no argument as in this case the string is split using spaces.

Skip Continue

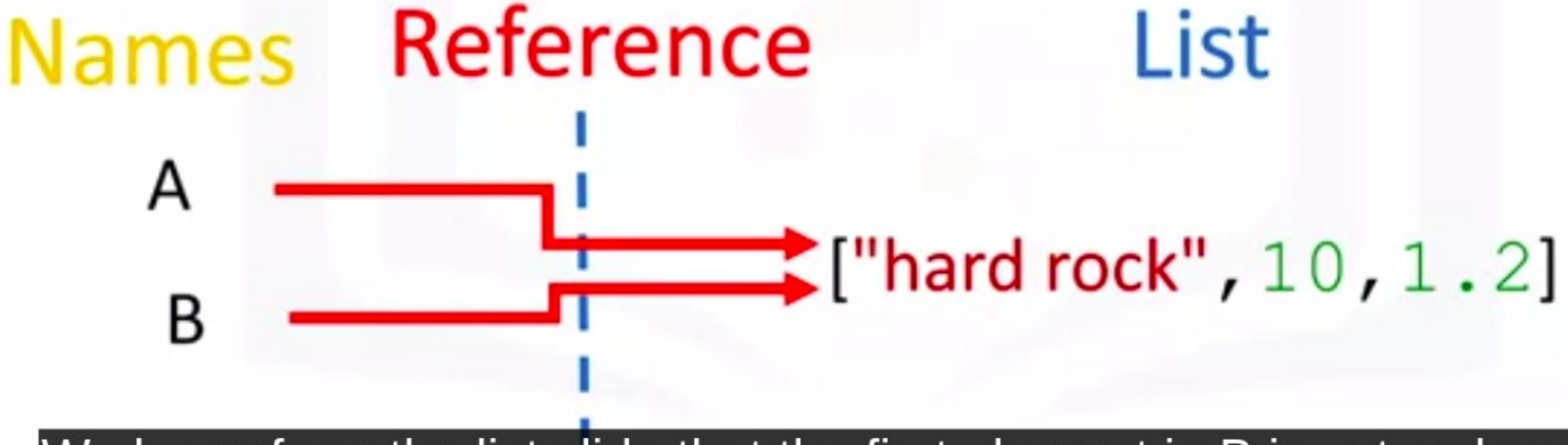
Lists



The result is a list. Each element corresponds to a set of characters that

Lists: Aliasing

B[0]="hard rock"

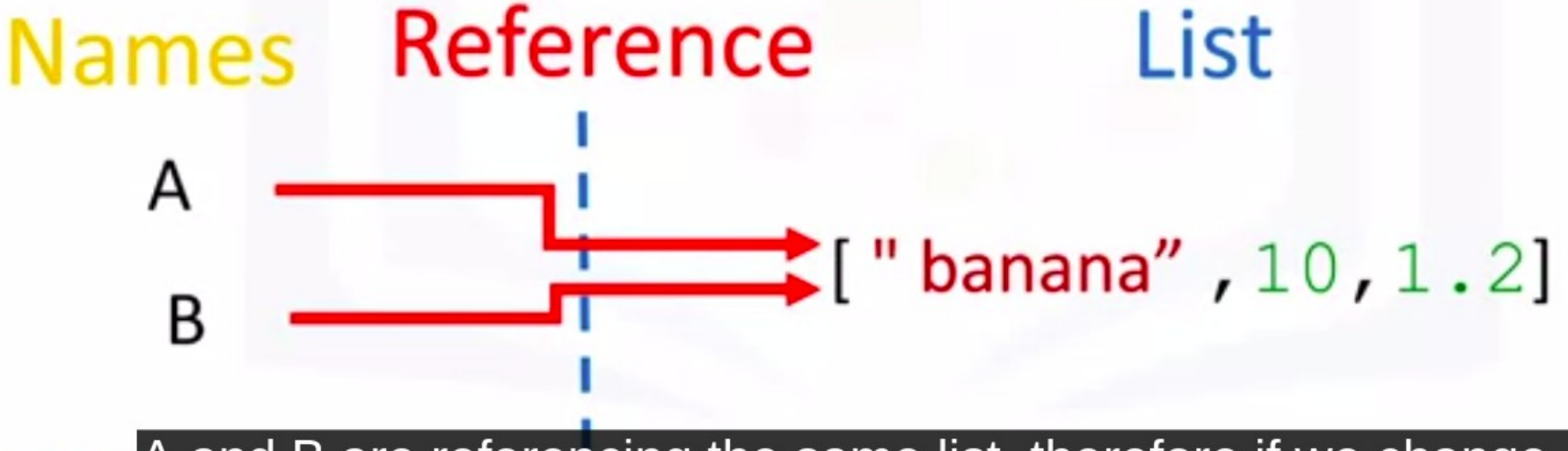


We know from the list slide that the first element in B is set as hard rock.

Lists: Aliasing

B[0]="hard rock”

A[0]=" banana ”

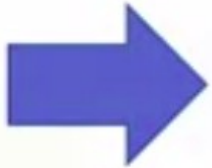


A and B are referencing the same list, therefore if we change A,

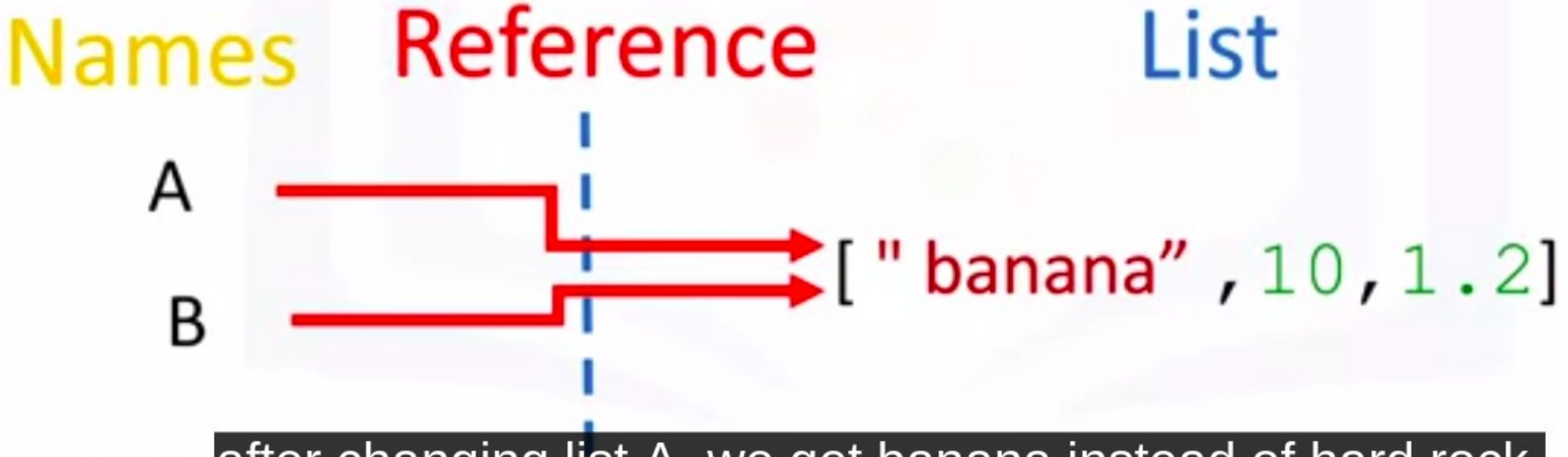
Lists: Aliasing

B[0]="hard rock"

A[0]=" banana "



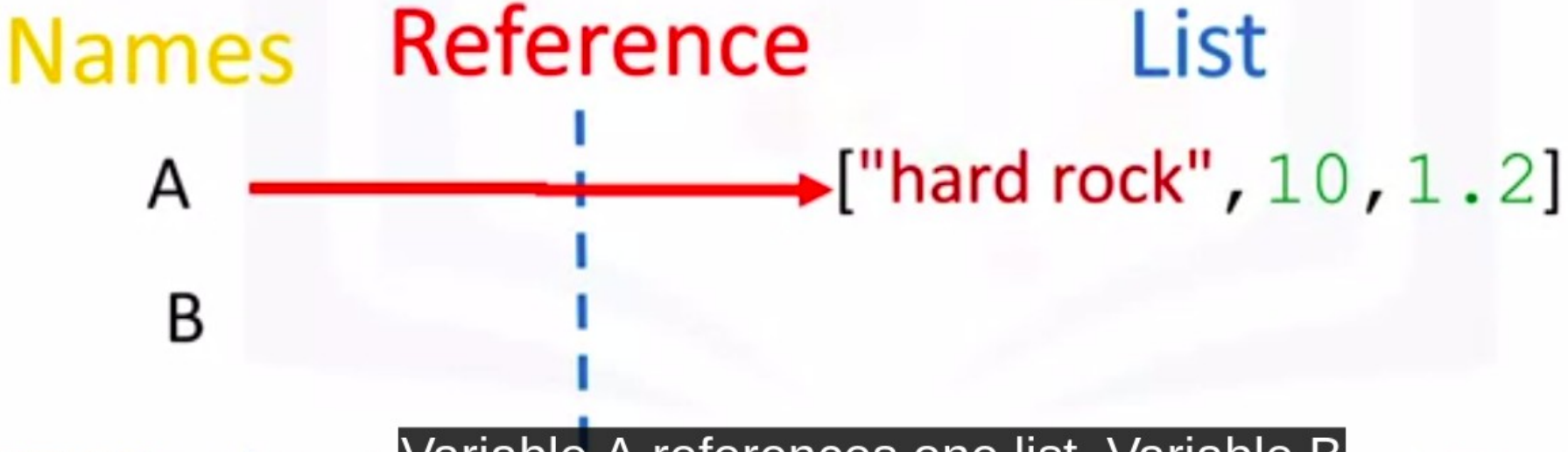
B[0]: " banana "



after changing list A, we get banana instead of hard rock.

Lists: Clone

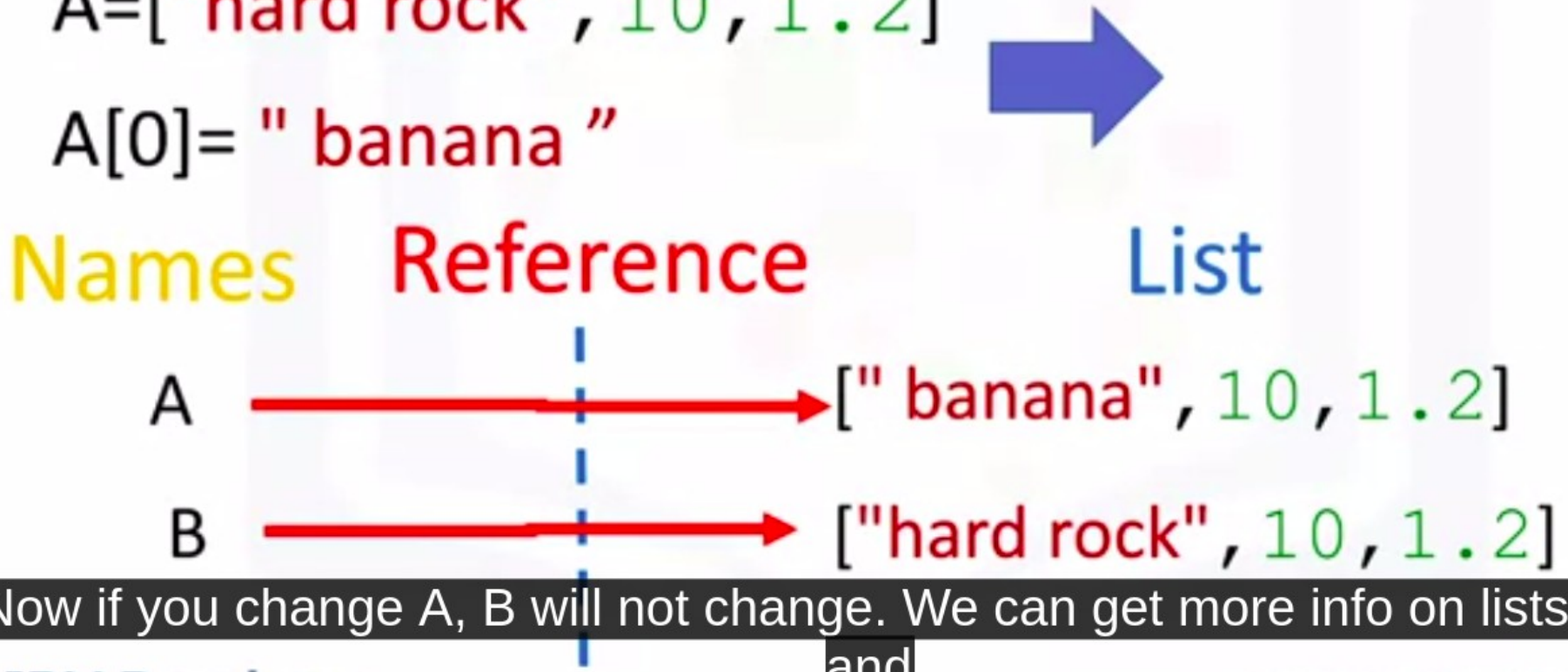
```
A=["hard rock",10,1.2]
B=A[:]
```



Variable A references one list. Variable B

Lists: Clone

```
A=["hard rock",10,1.2]
A[0]= " banana ”
```



Now if you change A, B will not change. We can get more info on lists, tuples, and

Back

Practice Quiz

Practice Quiz • 21 min • 7 total points

Congratulations! You passed!

Grade received 100% **To pass** 72% or higher

Go to next item

1. Consider the following tuple: **1 / 1 point**

say_what=('say',' what', 'you', 'will')

what is the result of the following **say_what[-1]**

- ( ) 'say'
- ( ) 'you'
- (•) 'will'
- ( ) ' what'

Correct

Correct. An index of -1 corresponds to the last index of the tuple, in this case, the string 'will'.

Back

Practice Quiz

Practice Quiz • 21 min • 7 total points

2. Consider the following tuple **A=(1,2,3,4,5).** What is the result of the following: **A[1:4]**: **1 / 1 point**

- (2, 3, 4)
- (3, 4,5)
- (2, 3, 4,5)

Correct

Correct. These indexes correspond to **elements** 1,2 and 3 of the tuple.

3. Consider the following tuple **A=(1,2,3,4,5)**, what is the result of the following: **len(A)** **1 / 1 point**

- 5
- 6
- 4

Correct

Correct. The function **len** returns the number of items of a tuple.

Back

Practice Quiz

Practice Quiz • 21 min • 7 total points

4. Consider the following list **B=[1,2,[3,'a'],[4,'b']]**, what is the result of the following:**B[3][1]** **1 / 1 point**

- (●) "b"
- ( ) [4,"b"]
- ( ) "c"

Correct
Correct.

5. What is the result of the following operation? **1 / 1 point**

[1,2,3]+[1,1,1]

- (●) [1, 2, 3, 1, 1, 1]
- ( ) [2,3,4]
- ( ) TypeError

Correct

Back

Practice Quiz

Practice Quiz • 21 min • 7 total points

6. What is the length of the list **A = [1]** after the following operation: **A.append([2,3,4,5])** **1 / 1 point**

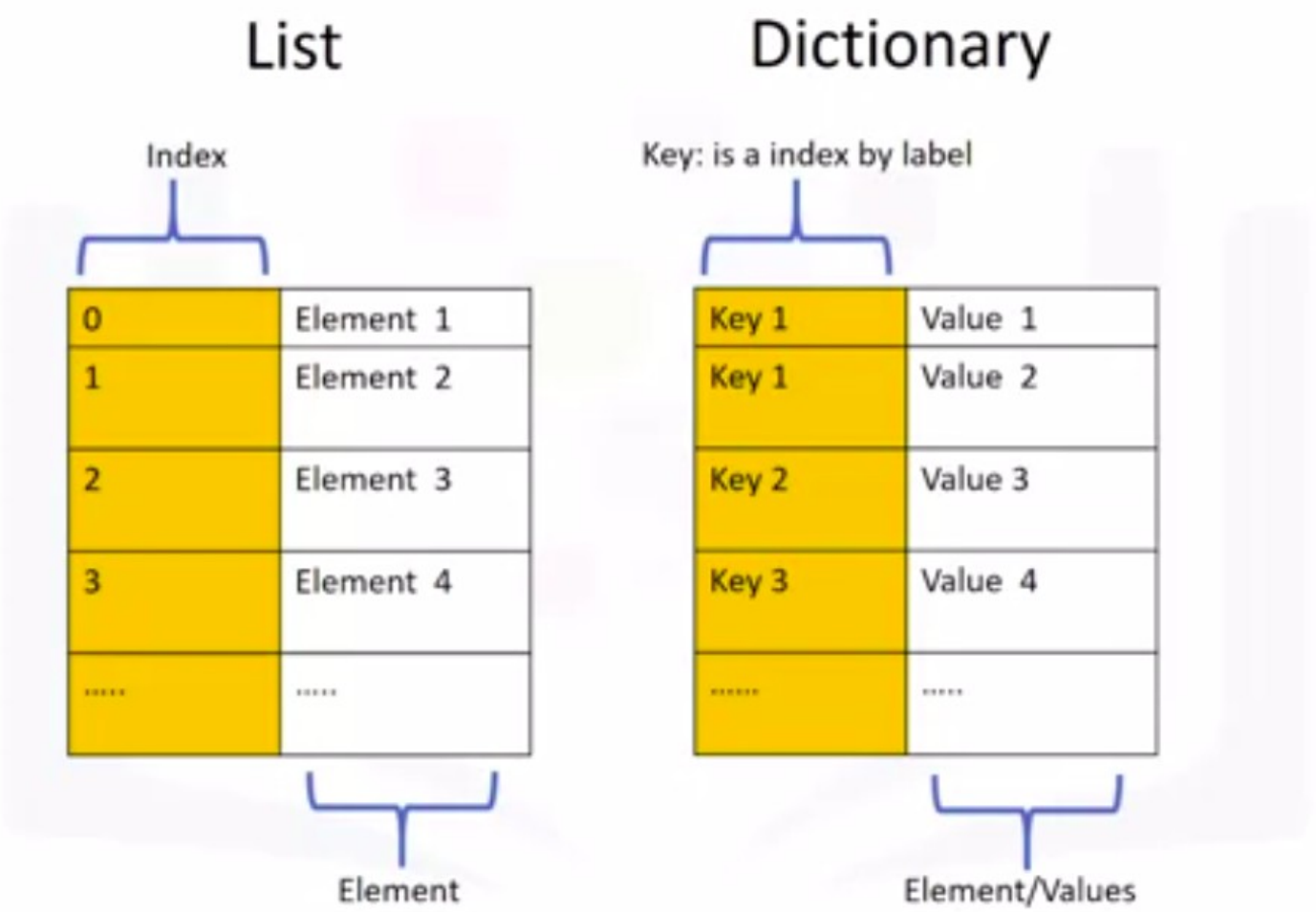
- (•) 2
- ( ) 5
- ( ) 6

Correct
Correct. Append only adds one element to the list .

7. What is the result of the following: **"Hello Mike".split()** **1 / 1 point**

- ( ) ["HelloMike"]
- (•) ["Hello","Mike"]
- ( ) ["H"]

Correct
Correct. The method split separates a string into a list based on the argument. If there is no argument as in this case the string is split using spaces.



The values are similar to the element in a list and contain information.

Dictionaries

- Dictionaries are denoted with curly Brackets {}
- The keys have to be immutable and unique
- The values can be can immutable, mutable and duplicates
- Each key and value pair is separated by a comma

{“key1”:1, "key2 ” :"2","key3” :[3,3,3], "key4":(4,4,4), ('key5’):5}

Each key and value pair is separated by a comma.

Question

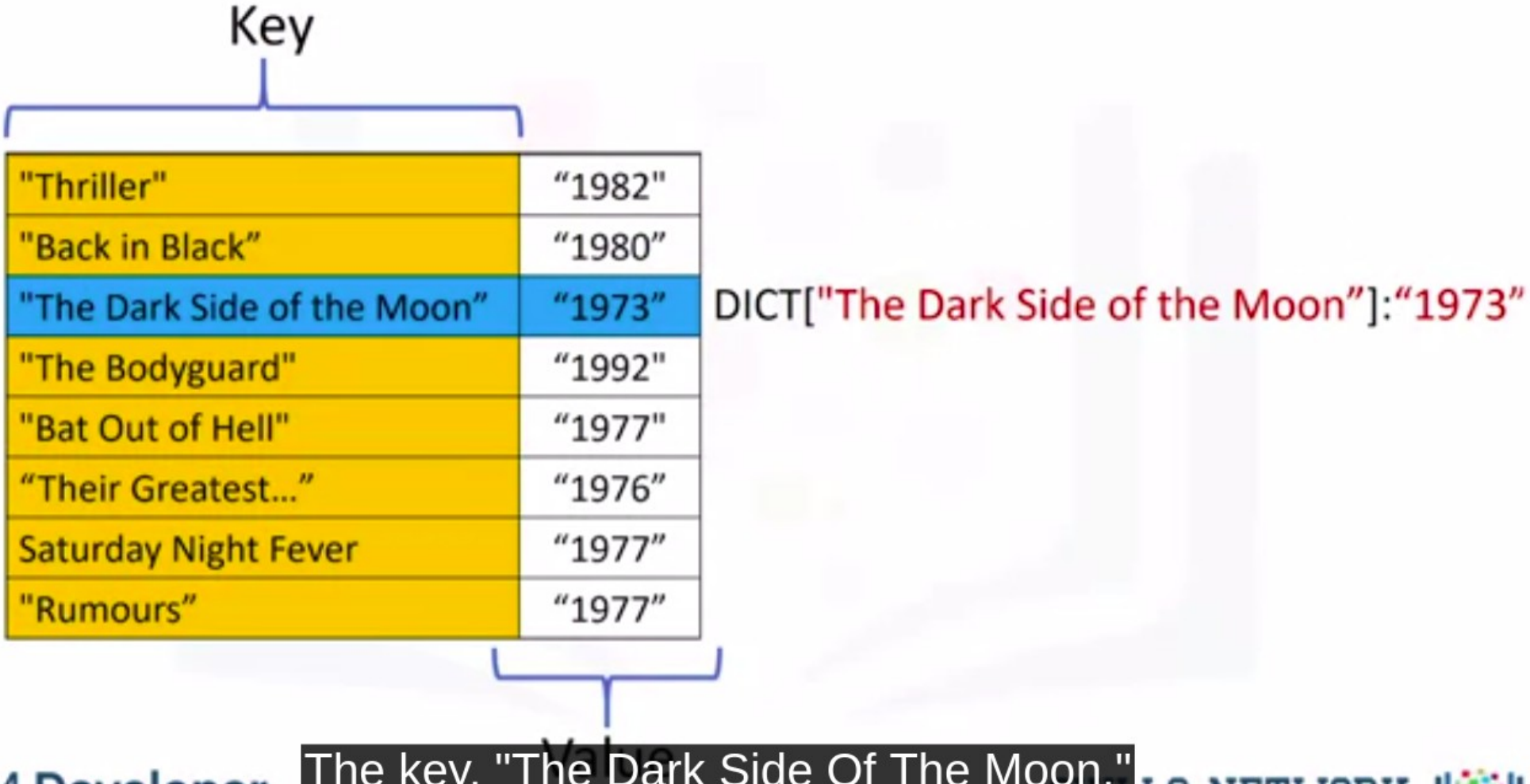
what are the keys of the following dictionary: **{"a":1,"b":2}**

- (●) "a","b"
- ( ) 1,2

✓ **Correct**

correct, the key is the first element separated from its value by a colon

Skip Continue



The key, "The Dark Side Of The Moon,"

DICT[“The Bodyguard”]:“1992”

Using the key,"The bodyguard,"

Question

Consider the following Python Dictionary:

Dict={"A":1,"B":"2","C":[3,3,3],"D":(4,4,4),'E':5,'F':6}

What is the result of the following operation: **Dict["D"]**

- [ ] [3,3,3]
- [x] (4, 4, 4)
- [ ] 1

Correct

correct, this corresponds to the key **'D'** or **Dict['D']**

Skip Continue

"Back in Black”	“1980”
"The Dark Side of the Moon”	“1973”
"The Bodyguard"	“1992"
"Bat Out of Hell"	“1977"
“Their Greatest...”	“1976”
Saturday Night Fever	“1977”
"Rumors”	“1977”

```
del(DICT['Thriller'])
```

"Thriller" and it's value.

"Thriller"	“1982"
"Back in Black”	“1980”
"The Dark Side of the Moon”	“1973”
"The Bodyguard"	“1992"
"Bat Out of Hell"	“1977"
“Their Greatest…”	“1976”
Saturday Night Fever	“1977”
"Rumors”	“1977”

'The Bodyguard' in DICT

If they are in the dictionary,

"Thriller"	“1982"
"Back in Black”	“1980”
"The Dark Side of the Moon”	“1973”
"The Bodyguard"	“1992"
"Bat Out of Hell"	“1977"
“Their Greatest…”	“1976”
Saturday Night Fever	“1977”
"Rumors”	“1977”

“Starboy” **in** DICT

False

we get a false.

"Thriller"	“1982"
"Back in Black”	“1980”
"The Dark Side of the Moon”	“1973”
"The Bodyguard"	“1992"
"Bat Out of Hell"	“1977"
“Their Greatest…”	“1976”
“Saturday Night Fever”	“1977”
"Rumors”	“1977”

DICT.keys()=["Thriller”, "Back in Black”, "The Dark Side of the Moon”, "The Bodyguard”, "Bat Out of Hell”, “Their Greatest…”,“Saturday Night Fever”, "Rumors”]

The output is a list-like object with all the keys.

"Thriller"	“1982"
"Back in Black”	“1980”
"The Dark Side of the Moon”	“1973”
"The Bodyguard"	“1992"
"Bat Out of Hell"	“1977"
“Their Greatest…”	“1976”
Saturday Night Fever	“1977”
"Rumors”	“1977”

DICT.values() =[“1982”,“1980”,”1973”,“1992”, “1977”,“1976” “1977”, “1977”]

In the same way, we can obtain the values using the method values.

Back

Practice Quiz

Practice Quiz • 6 min • 2 total points

1. What are the keys of the following dictionary: **{"a":1,"b":2}** **1 / 1 point**

- (●) "a","b"
- ( ) 1,2

Correct
Correct, the key is the first element separated from its value by a colon.

2. Consider the following Python Dictionary: **1 / 1 point**

Dict={"A":1,"B":"2","C":[3,3,3],"D":(4,4,4),'E':5,'F':6}

What is the result of the following operation: **Dict["D"]**

- ( ) [3,3,3]
- (●) (4, 4, 4)
- ( ) 1

Correct
Correct, this corresponds to the key **'D'** or **Dict['D']**

Sets

- Sets are a type of collection
  - This means that like lists and tuples you can input different Python types
- Unlike lists and tuples they are unordered
  - This means sets do not record element position
- Sets only have unique elements
  - This means there is only one of a particular element in a set

This means there is only one of a particular element in a set.

Question

Consider the following set: **{"A","A"}**, what will the result be when the set is created

- ○ {"A","A"}
- ◉ {"A"}

✓ **Correct**
correct, there are no duplicate values in a set

Skip Continue

Sets: Creating a Set

```
album_list = ["Michael Jackson", "Thriller", "Thriller", 1982]

album_set = set(album_list)

album_set :  {'Michael Jackson', 'Thriller', 1982}
```

set()

album_set

The function set returns a set.

Sets: Creating a Set

```
album_list = ["Michael Jackson", "Thriller", "Thriller", 1982]

album_set = set(album_list)

album_set :  {'Michael Jackson', 'Thriller', 1982}
```

set()

album_set

Notice how there are no duplicate elements.

Question

What is the result of the following: **type(set([1,2,3]))**

- (•) set
- ( ) list

✓ **Correct**
correct, the function **set** casts the list to a set before we apply the **type** function

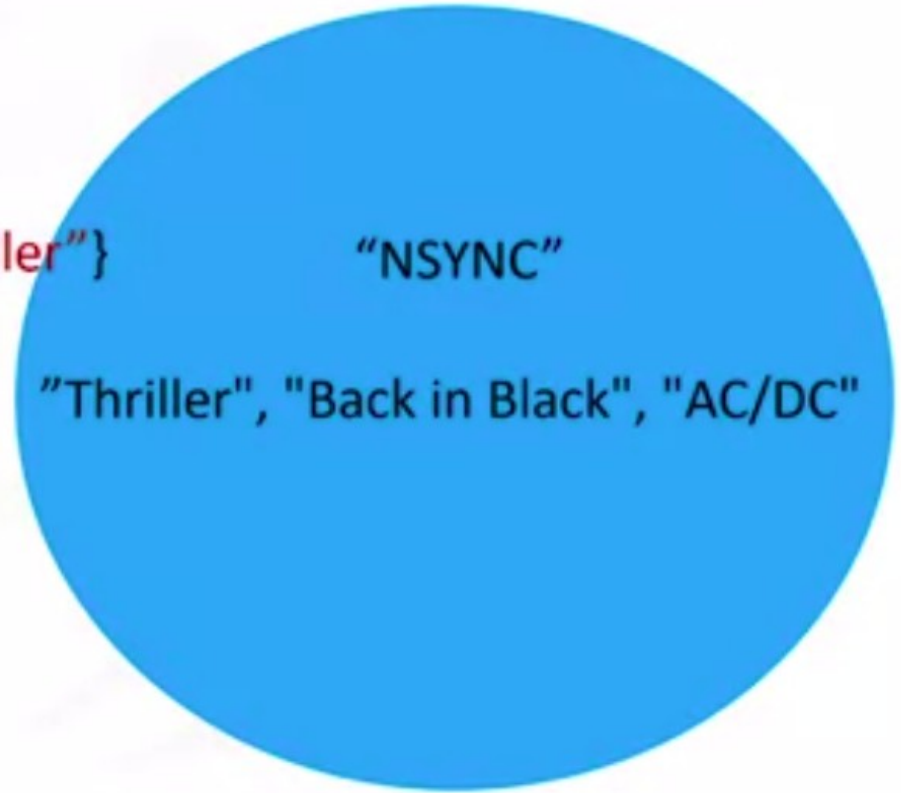
Skip Continue

Set Operations

A = {“Thriller”, “Back in Black”, “AC/DC”}

A.add("NSYNC")

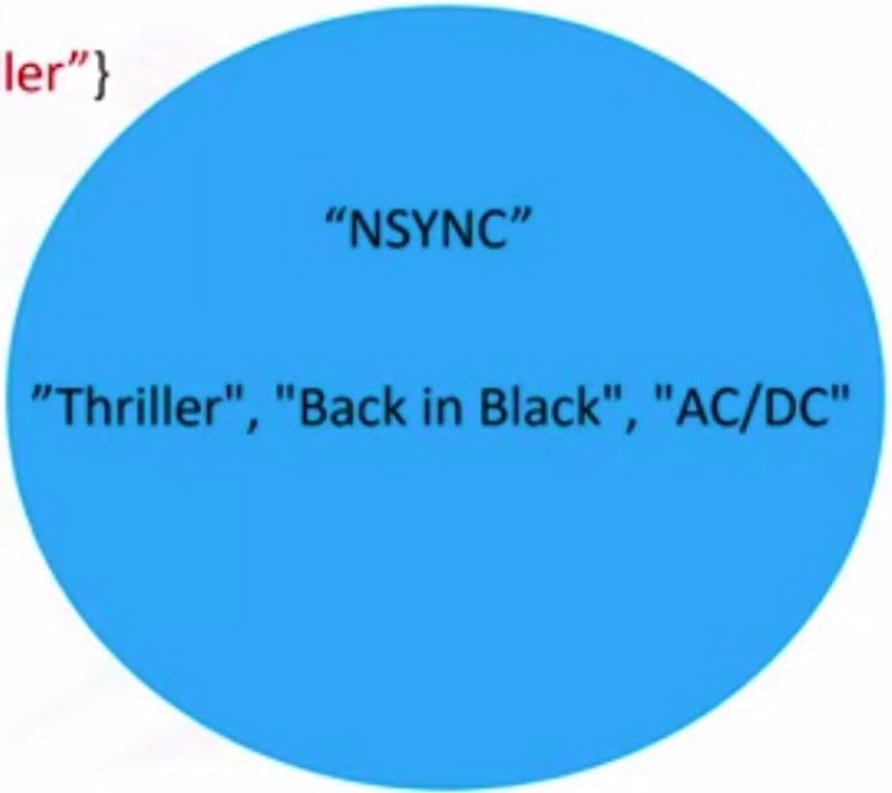
A:{“AC/DC”, “Back in Black”, “NSYNC”, “Thriller”}



The set A now has in NSYNC as an item.

Set Operations

A :{“AC/DC”, “Back in Black”, “NSYNC”, “Thriller”}



Let's say we would like to remove NSYNC from set A.

Question

what method do you use to add an element to a set

- ○ append
- ○ extend
- ◉ add

✓ **Correct**
correct

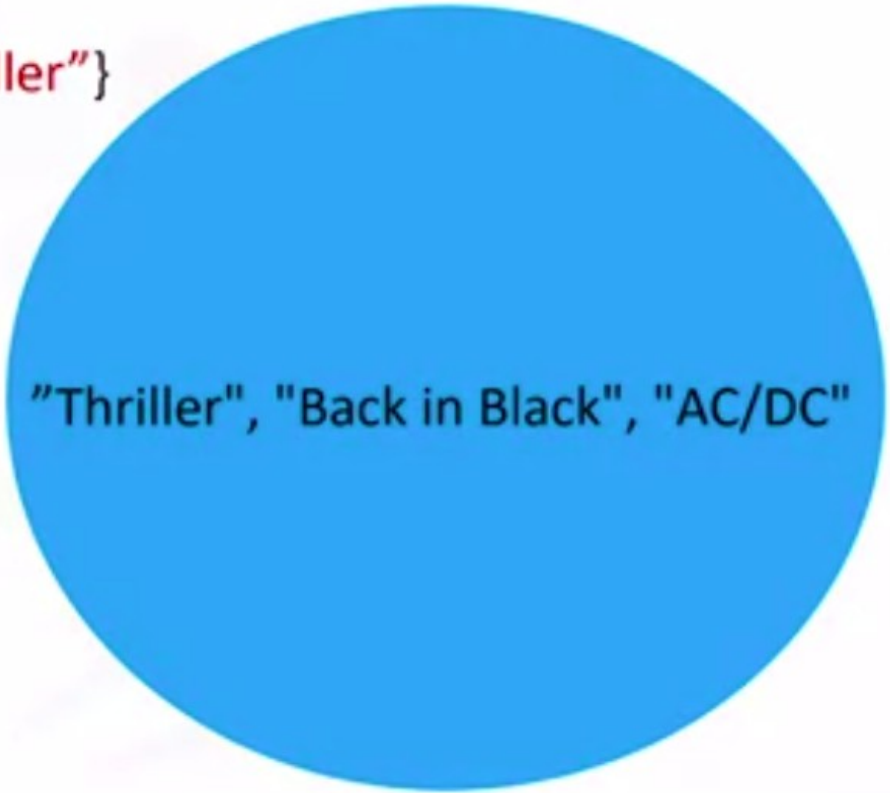
Skip Continue

Set Operations

A :{“AC/DC”, “Back in Black”, “NSYNC”, “Thriller”}

A.remove("NSYNC")

A:{“AC/DC”, “Back in Black”, “Thriller”}



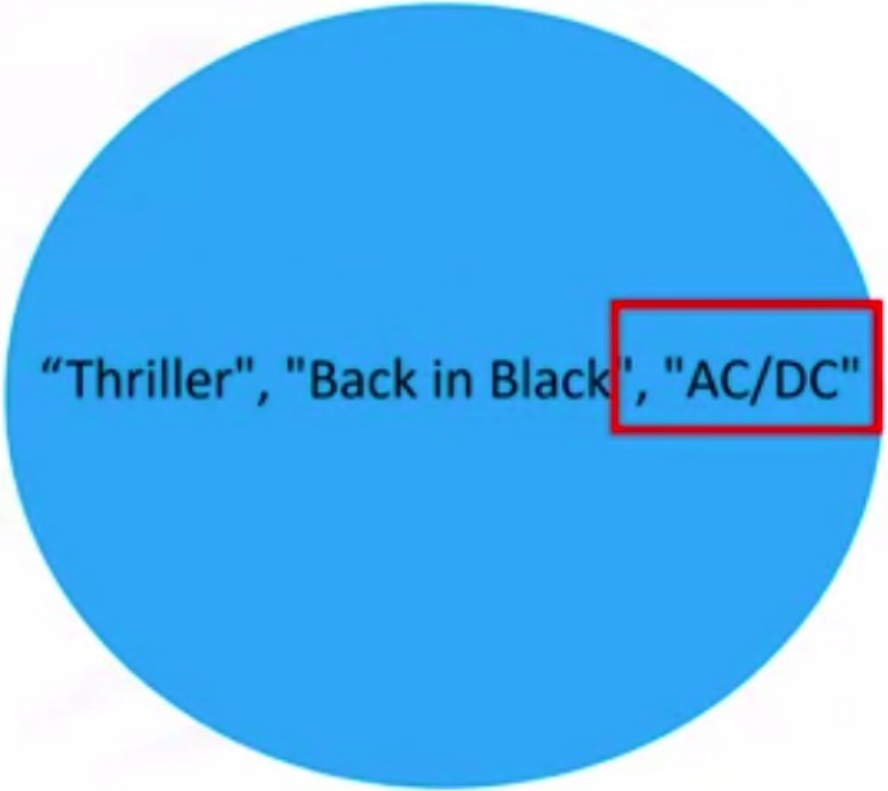
The argument is the element of the set we would like to remove,

Set Operations

A:{“AC/DC”, “Back in Black”, “Thriller”}

“AC/DC” in A

True



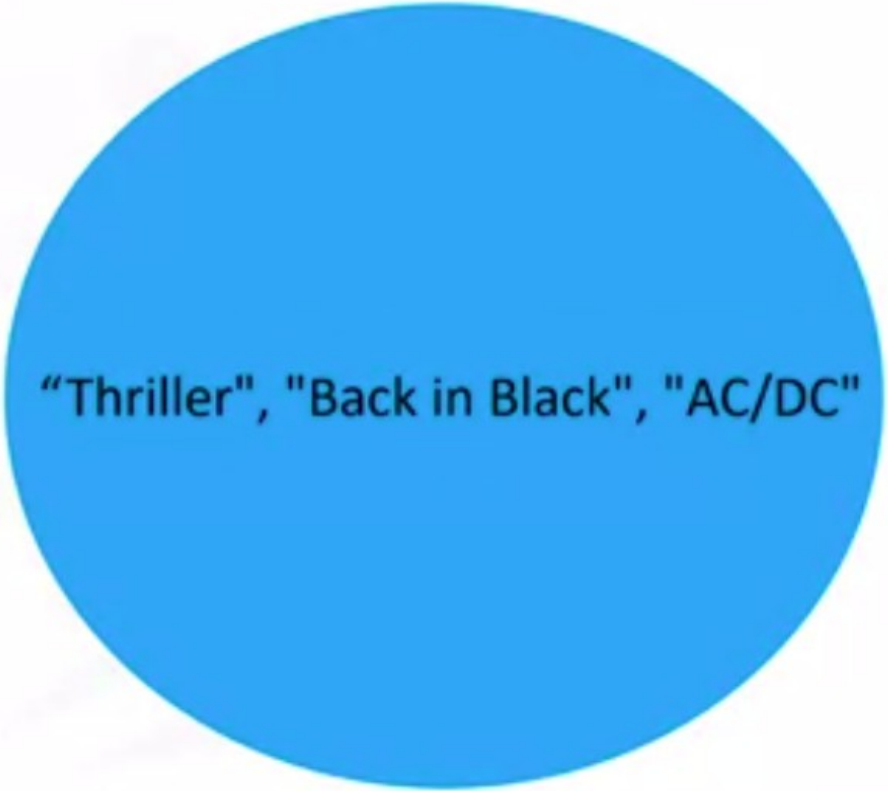
If the item is in the set, it returns true.

Set Operations

A:{“AC/DC”, “Back in Black”, “Thriller”}

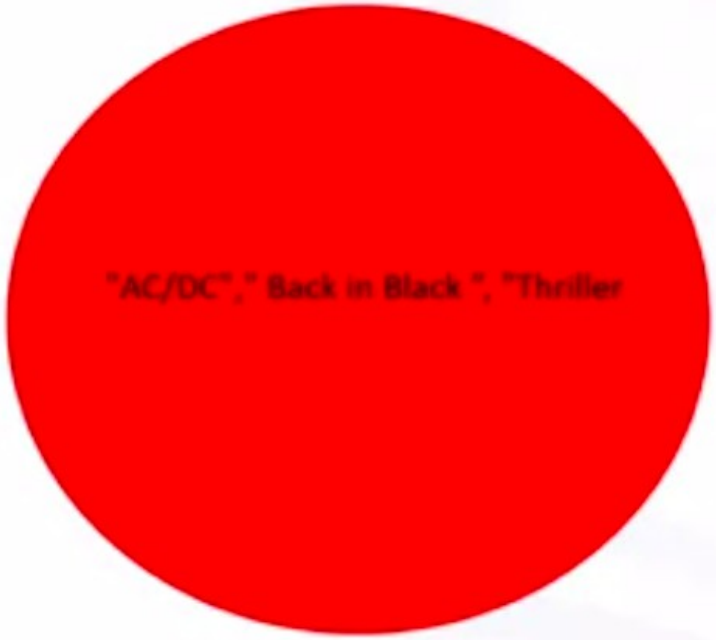
“Who” in A

False



adds the item is not in the set,

Sets: Mathematical set operations



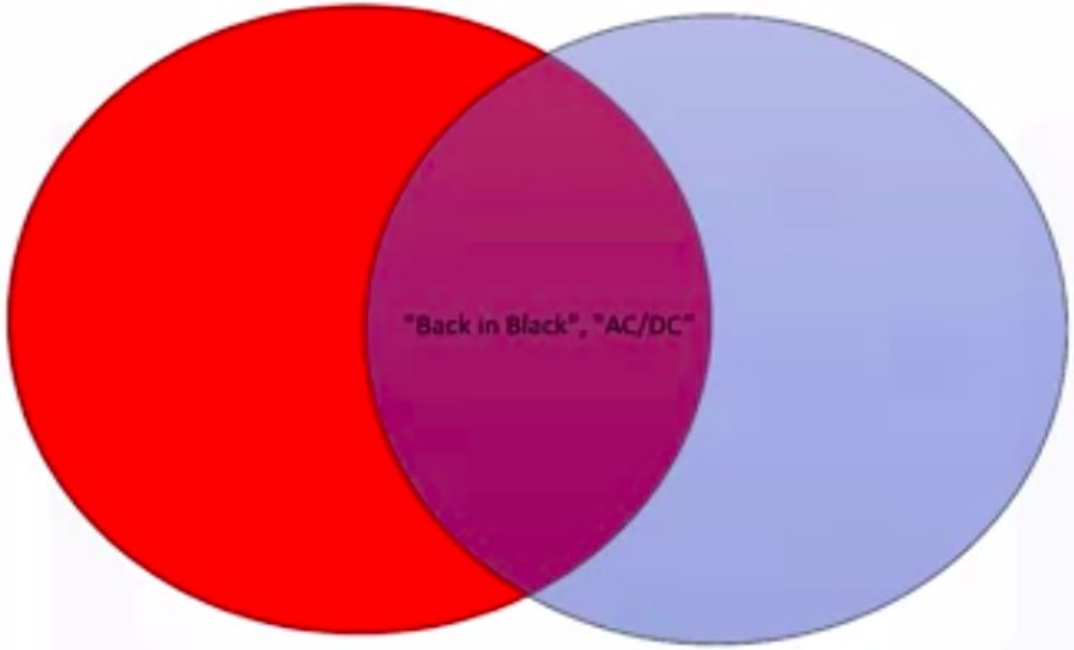
album_set_1 = {"AC/DC"," Back in Black ", "Thriller }

We can represent it using a red circle or venn diagram.

Sets: Mathematical set operations

album_set_2 = {"AC/DC","Back in Black", "The Dark Side of the Moon"}

We can also represent it using a blue circle or venn diagram.



all the items that are not in both sets disappear.

album_set_1 ={"AC/DC"," Back in Black ", "Thriller }

album_set_2={"AC/DC", "Back in Black", "The Dark Side of the Moon"}

album_set_3=album_set_1 & album_set_2

album_set_3: {"AC/DC"," Back in Black "}

The result is a new set album: set three

Question

what is the result of the following operation : **{'a','b'} &{'a'}**

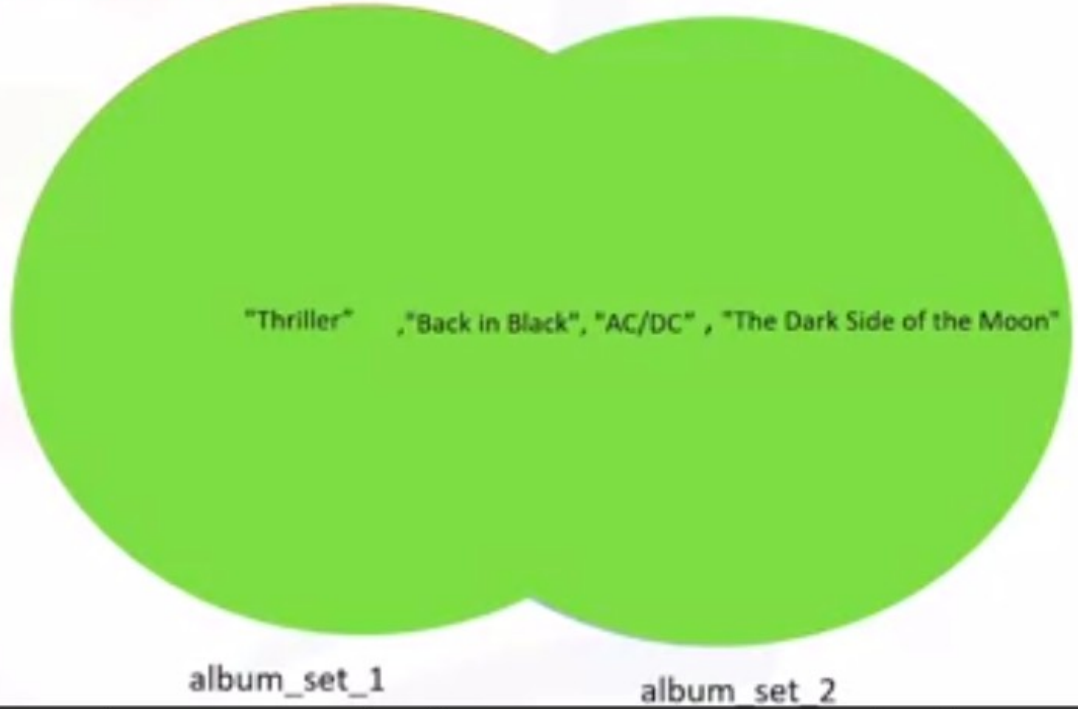
- (•) {'a'}
- ( ) {'a','b'}

Correct
correct, the intersection operation finds the elements that are in both sets

Skip Continue

album_set_1. union(album_set_2)

{AC/DC", "Back in Black", "The Dark Side of the Moon", "Thriller" }



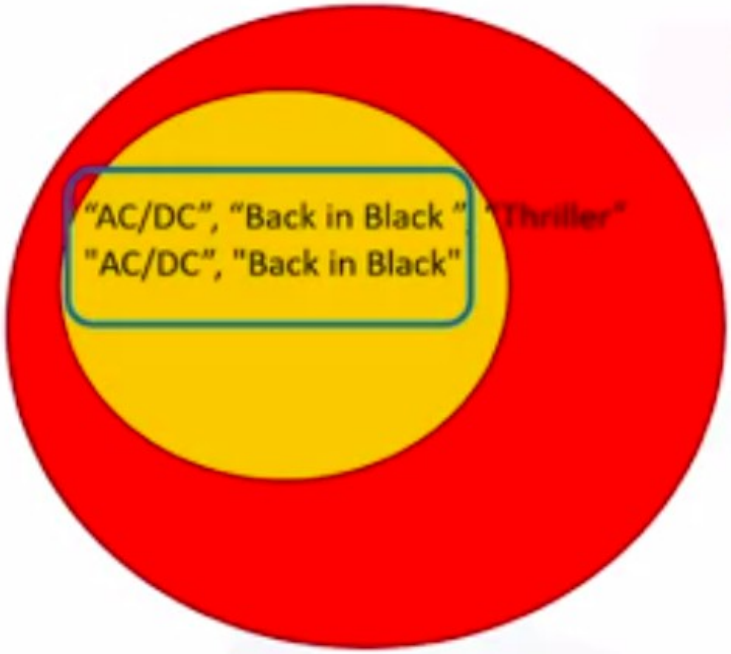
The result is a new set that has all the elements of album set one and album set two.

As album set three is a subset of the album set one,

album_set_1 = {"AC/DC"," Back in Black ", "Thriller }
album_set_3 = {"AC/DC"," Back in Black "}

album_set_3.issubset(album_set1)

True



As album set three is a subset of the album set one,

Back

Practice Quiz

Practice Quiz • 12 min • 4 total points

1. Consider the following set: **{"A","A"}**, what will the result be when the set is created? **1 / 1 point**

- (●) {"A"}
- ( ) {"A","A"}

Correct
correct, there are no duplicate values in a set

2. What is the result of the following: **type(set([1,2,3]))** **1 / 1 point**

- (●) set
- ( ) list

Correct
correct, the function **set** casts the list to a set before we apply the **type** function

Back

Practice Quiz

Practice Quiz • 12 min • 4 total points

3. What method do you use to add an element to a set? **1 / 1 point**

- ○ extend
- ◉ add
- ○ append

Correct
correct

4. What is the result of the following operation : **{'a','b'} &{'a'}** **1 / 1 point**

- ◉ {'a'}
- ○ {'a','b'}

Correct
Correct, the intersection operation finds the elements that are in both sets.

Back

Module 2 Graded Quiz

Graded Quiz • 30 min

Due Aug 21, 11:59 PM IST

1. Consider the tuple **A=((11,12),[21,22])**, that contains a tuple and list. What is the result of the following operation **A[1]** ? **1 / 1 point**

- ((11,12),[21,22])
- (11,12)
- [21,22]

Correct
correct, the index 1 corresponds to the second element in the tuple, which contains another list.

2. Consider the tuple **A=((11,12),[21,22])**, that contains a tuple and list. What is the result of the following operation **A[0][1]**? **1 / 1 point**

- 21
- 11
- 12

Correct
correct, A[0] corresponds to the first nested tuple; we then access the second element of the tuple using the index 1 i.e A[0][1].

3. The method append does the following: **1 / 1 point**

- (●) adds one element to a list
- ( ) merges two lists or insert multiple elements to a list

Correct
correct, append-only adds one element.

4. Consider the following list : **A=["hard rock",10,1.2]** **1 / 1 point**

What will list **A** contain after the following command is run: **del(A[0])** ?

- (●) [10,1.2]
- ( ) ["hard rock",10,1.2]
- ( ) ["hard rock",10]

Correct
correct, we will delete element zero

5. What is the syntax to clone the list **A** and assign the result to list **B** ? **1 / 1 point**

- ○ **B=A**
- ◉ **B=A[:]**

Correct
correct

6. What is the result of the following: **len(("disco",10,1.2, "hard rock",10))** ? **1 / 1 point**

- ◉ 5
- ○ 6
- ○ 0

Correct
correct, there are 5 elements in the tuple so the function len returns 5

7. Consider the following dictionary: 1 / 1 point

{ "The Bodyguard":"1992", "Saturday Night Fever":"1977"}

select the keys

- [ ] "1992"
- [ ] "1977"
- [x] "Saturday Night Fever"

Correct
correct, this is one of the keys

- [x] "The Bodyguard"

Correct
correct, this is one of the keys

8. The variable **release_year_dict** is a Python Dictionary, what is the result of applying the following method: **release_year_dict.values()** ? 1 / 1 point

8. The variable **release_year_dict** is a Python Dictionary, what is the result of applying the following method: **release_year_dict.values()** ? **1 / 1 point**

- ○ retrieve the keys of the dictionary
- ◉ retrieves, the values of the dictionary

Correct
correct, this method returns the values

9. Consider the Set: **V={'A','B'}**, what is the result of **V.add('C')**? **1 / 1 point**

- ○ {'A','B'}
- ◉ {'A','B','C'}
- ○ error

Correct
correct

9. Consider the Set: **V={'A','B'}**, what is the result of **V.add('C')**? **1 / 1 point**

- ○ {'A','B'}
- ◉ {'A','B','C'}
- ○ error

Correct
correct

10. What is the result of the following: **'A' in {'A','B'}** ? **1 / 1 point**

- ◉ True
- ○ False

Correct
correct